सी-डैक
CDAC
CENTER FOR DEVELOPMENT OF
ADVANCED COMPUTING

Project Report

On

DISASTROUS TWEETS WITH NLP (NATURAL LANGUAGE PROCESSING)

Report submitted to
Centre for Development of advanced Computing
for the award of
Post Graduate
Diploma in Big Data Analytics from **C-DAC ACTS, PUNE**
Batch Sept 2020-March 2021

under the guidance of

Mr Prateek Maheshwari

Ms Seema Sajeevan (Course coordinator)	Ms Risha P.R (Manager ACTS)

Presented by: -

Devina Yadav	**PRN-200940183011**
Eniya Kulshrestha	**PRN-200940183014**
Samridhi Pandey	**PRN-200940183044**
Shrishti Verma	**PRN-200940183048**

CERTIFICATE

TO WHOMSOEVER IT MAY CONCERN

This is to certify that the Project Report entitled, "Disastrous tweets with NLP (Natural Language Processing)" submitted by Group-10 to CDAC-ACTS, Pune is a record of bonafide Project work carried out by them under our supervision and guidance and is worthy of consideration for the award of the Post Graduate Diploma in Big Data Analytics (PG-DBDA) from CDAC-ACTS, Pune (Maharashtra).

----------------------------------   --------------------------------

Project Guide   Course Coordinator

(Digital Signature)   (Digital Signature)

HOD

(Digital Signature)

Acknowledgement

This project "Disastrous Tweets with NLP(Natural Language Processing)" was a great learning experience for us and we are submitting this work to CDAC-ACTS, Pune

We all are very glad to mention the name of Mr Prateek Maheswari for his valuable guidance to work on this project. His guidance and support helped us a lot to overcome various obstacles and core intricacies during the course of project work.

We are highly grateful to Ms. Risha P.R. (Manager (ACTS training Centre), C-DAC), for her guidance and support whenever necessary while doing this course Post Graduate Diploma in *Big Data Analytics (PGDBDA)* through C-DAC ACTS, Pune.

Our most heartfelt thanks goes to *Ms. Seema Sajeevan* (Course Coordinator, *PGDBDA*) who gave all the required support and kind coordination to provide all the necessities like required help and extra Lab hours to complete the project and throughout the course up to the last day here in C-DAC ACTS, Pune

DECLARATION

We certify that the work contained in this report is original and has been done by us under the guidance of our supervisor. The work has not been submitted to any other Institute for any degree or diploma as we have followed the guidelines provided by the Institute in preparing the report. Whenever we have used materials (data, theoretical analysis, figures, and text) from other sources, we have given due credit to them by citing them in the text of the report and giving their details in the references. Further, we have taken permission from the copyright owners of the sources, whenever necessary.

Signature of the Students

CONTENTS

1. ABSTRACT

Social networks offer a wealth of information during any human-affecting events such as natural disasters. In this research, we use natural language processing to predict whether or not text is referring to a natural disaster. Using the data, consisting of 10,000 labelled Tweets, we apply machine learning models to determine which Tweets are about real disasters and which ones are not.

We first perform some exploratory data analysis, visualizing the dataset, and then text pre-processing to remove irrelevant characters. We discover that most disasters are related to natural disasters and criminal activity, while non-disasters, focus on slang and YouTube videos. We implement five machine learning models: Naive Bayes, Support Vector Machines, Random Forest, Logistic Regression Logistic Regression appear to have slightly more success than the other models as the accuracy for these is about 81%. Finally, we tune the model parameters so as to increase the performance of our predictions. We have also proposed a natural-disaster analysis interface that solely makes use of tweets generated by the Twitter users. We collect streaming tweets relating to disasters and the results are presented in the form of detailed graphical analysis which demonstrates disastrous and non-disastrous tweet.

2. Introduction and Overview of Project

Twitter has become an important communication channel in times of emergency.
The ubiquitousness of smartphones enables people to announce an emergency they're observing in real-time. Because of this, more agencies are interested in programmatically monitoring Twitter (i.e. disaster relief organizations and news agencies and government).

But, it's not always clear whether a person's words are actually announcing a disaster.

Data-driven models are based on a strictly mathematical model and measurements.

Consequently, they do not require such detailed knowledge of the building or equipment. Their forecasts are mostly based on historical data which are more readily available from control systems implemented within the building.

The accuracy of these models, when applied to forecasting, depends on the quality of the selected forecasting model, the quality, and quantity of data available. Such models are easily adaptable to changing conditions, can model nonlinear phenomena, and are relatively easy to train and use. In most cases, the relationship between the forecasted variable and its driving physical functions is not explicitly derived.

3.Objectives of The Project:

Objective of this capstone project is to understand the business domain, analyse and explore the given data set then train and fit the machine learning model and finally classify given test tweets data into announcing disaster or not. Disaster relief organizations and News Agencies would be the main users of this kind of application.

NLP (Natural Language Processes sing) used: -

- Natural language processing is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyse large amounts of natural language data.
- NLP makes it possible for computers to read text, hear speech, interpret it, measure sentiment and determine which parts are important.

Models Used for Prediction in this project (best accuracy): -

1. Logistic Regression

2. Multinomial NB

3. Random Forest

4.SVM

EXPLANATION OF MODELS:

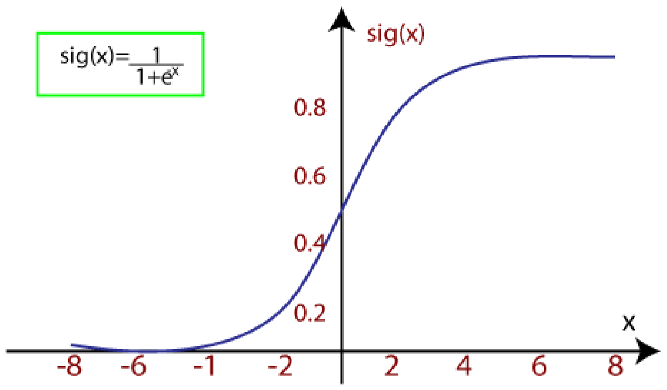
1) Logistic Regression Model: -

- *Logistic Regression is a mathematical model used in statistics to estimate (guess) the probability of an event occurring using some previous data. Logistic Regression works with binary data, where either the event happens (1) or the event does not happen (0).*
- Logistic regression algorithm works with the categorical variable such as 0 or 1, Yes or No, True or False, Spam or not spam, etc.
- It is a predictive analysis algorithm which works on the concept of probability.
- Logistic regression uses sigmoid function or logistic function which is a complex cost function. This sigmoid function is used to model the data in logistic regression. The function can be represented as:

$$f(x)=\frac{1}{1+e^{-x}}$$

- f(x)= Output between the 0 and 1 value.
- x= input to the function
- e= base of natural logarithm.

When we provide the input values (data) to the function, it gives the S-curve as follows:



- It uses the concept of threshold levels, values above the threshold level are rounded up to 1, and values below the threshold level are rounded up to 0.
- There are three types of logistic regression:

Binary (0/1, pass/fail)

Multi (cats, dogs, lions)

Ordinal (low, medium, high): it is in order form

2) Multinomial NB: -

- It is mainly used in text classification that includes a high-dimensional training dataset.

- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.
- Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

$$Posterior\ Probability = \frac{Conditional\ Probability * Prior\ Probability}{Predictor\ Prior\ Probability}$$

Working of Naïve Bayes' Classifier:

Working of Naïve Bayes' Classifier can be understood with the help of the below example:

Suppose we have a dataset of **DISASTER TWEETs** and corresponding target variable. So using this dataset we need to decide that whether the tweet is disastrous or not. So, to solve this problem, we need to follow the below steps:

1. Convert the given dataset into frequency tables.
2. Generate Likelihood table by finding the probabilities of given features.
3. Now, use Bayes theorem to calculate the posterior probability.

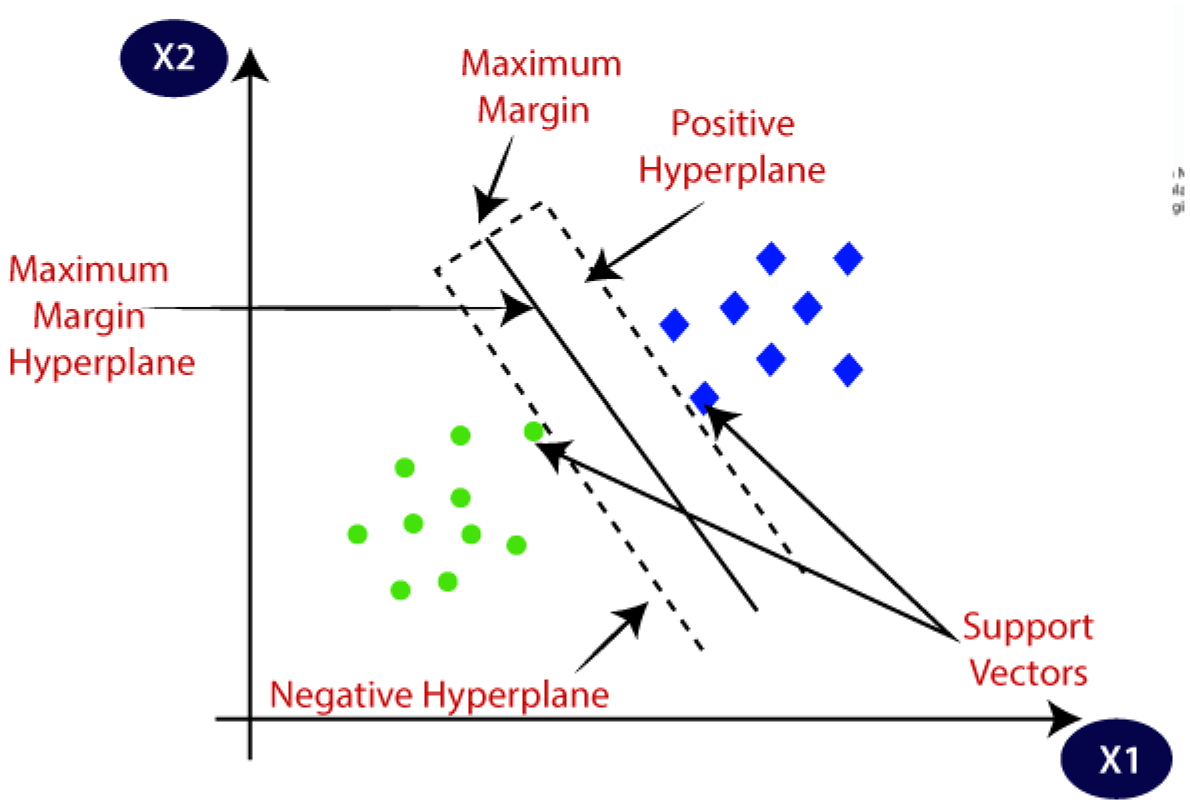
$$P\left(\frac{A}{B}\right) = \left(\frac{P(A \cap B)}{P(B)}\right) = \frac{P(A) * P\left(\frac{B}{A}\right)}{P(B)}$$

4. Random Forest: -

- "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.
- The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

5. SVM: -

- The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.
- SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.



4. Dataset Description

Data Exploration: - Given data set contains 2 files:

1. train.csv - the train dataset

2. test.csv - the test dataset

Each sample in the train and test set has the following information.

• The text of a tweet

• A keyword from that tweet (this may be blank)

• The location, the tweet was sent from (this may also be blank)

Features(columns) in the “train.csv” data file:

• id - a unique identifier for each tweet

• text - the text of the tweet

• location - the location the tweet was sent from (may be blank)

• keyword - a particular keyword from the tweet (may be blank)

• target - in train.csv only, where real disaster tweets are represented by “1” and non-disaster tweets are represented by “0”.

5. Data Pre-processing:

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data pre-processing task.

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

1. Data Cleaning

- Before starting any NLP project, text data needs to be pre-processed to convert it into in a consistent format. Text will be cleaned, tokenized and converted into a matrix.
- Some of the basic text per-processing techniques includes:

2. Make all text in lower/uppercase

- Algorithms does not treat the same word different in different cases

3. Removing Noise

- Everything in the text that isn't a standard number or letter i.e. Punctuation, Numerical values, URLs, special characters, etc.

4. Tokenization

- Tokenizing is the process of tokenizing or splitting a string, text into a list of tokens.
- Sentence tokenizer can be used to find the list of sentences and Word tokenizer can be used to find the list of words in strings.

To import: -

```
import nltk
from nltk. tokenize import word_tokenize
```

Key Points:

- Text into sentences tokenization

- Sentences into words tokenization
- Sentences using regular expressions tokenization

5. Stopwords Removal

- A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.
- We would not want these words to take up space in our database, or taking up valuable processing time. For this, we can remove them easily, by storing a list of words that you consider to stop words. NLTK (Natural Language Toolkit) in python has a list of stopwords stored in 16 different languages. You can find them in the nltk_data directory.

  To import:

```
import nltk
from nltk. corpus import stopwords
```

6. Stemming

- Stemming is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers. A stemming algorithm reduces the words “chocolates”, “chocolatey”, “Choco” to the root word, “chocolate” and “retrieval”, “retrieved”, “retrieves” reduce to the stem “retrieve”.
- Errors in Stemming:
- There are mainly two errors in stemming – Overstemming and Understemming. Overstemming occurs when two words are stemmed to same root that are of different stems. Under-stemming occurs when two words are stemmed to same root that are not of different stems.
  Applications of stemming are:
- Stemming is used in information retrieval systems like search engines.
- It is used to determine domain vocabularies in domain analysis.

7. Lemmatization

- A slight variant of stemming is lemmatization. The major difference between these is, that, stemming can often create non-existent words, whereas lemmas are actual words. So, your root stem, meaning the word you end up with, is not something you can just look up in a dictionary, but you can look up a lemma. Examples of Lemmatization are that “run” is a base form for words like “running” or “ran” or that the word “better” and “good” are in the same lemma so they are considered the same.

Applications of lemmatization are:

- Used in comprehensive retrieval systems like search engines.
- Used in compact indexing

8. Vectorization of pre-processed text

- In Pre-processed text needs to be transformed into a vector matrix of numbers before a machine learning model can understand it and learn from it. This can be done by a number of techniques:

TF-IDF algorithm

- A problem with the bag of words approach is that highly frequent words start to dominate in the document (e.g. larger score) but may not contain as much "informational content" this will lead to more weight to longer documents than shorter documents.
- TF-IDF is applied on the body text, so the relative count of each word in the sentences is stored in the document matrix
- To avoid that, one approach is to re-scale the frequency of words by how often they appear in all documents so that the scores for frequent words like "the" that are also frequent across all documents are penalized. This approach to scoring is called "Term Frequency-Inverse Document Frequency", or TF-IDF for short, where:

• Term Frequency: is a scoring of the frequency of the word in the current document.

TF = (Number of times term t appears in a document)/ (Number of terms in the document)

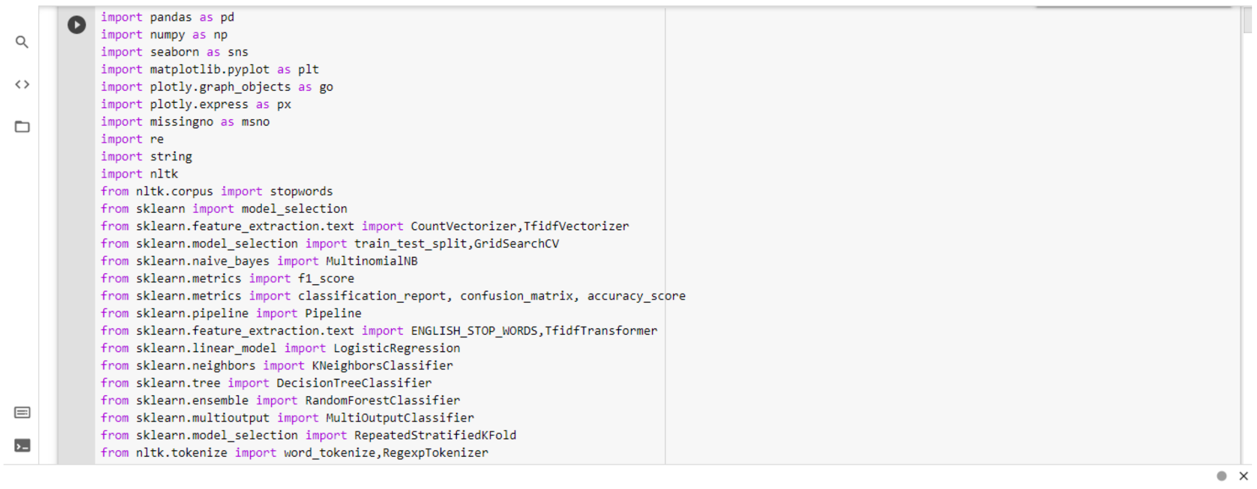
• Inverse Document Frequency: is a scoring of how rare the word is across documents.

IDF = 1+log(N/n), where, N is the number of documents and n is the number of documents a term t has appeared in.

6. Exploratory Data Analysis

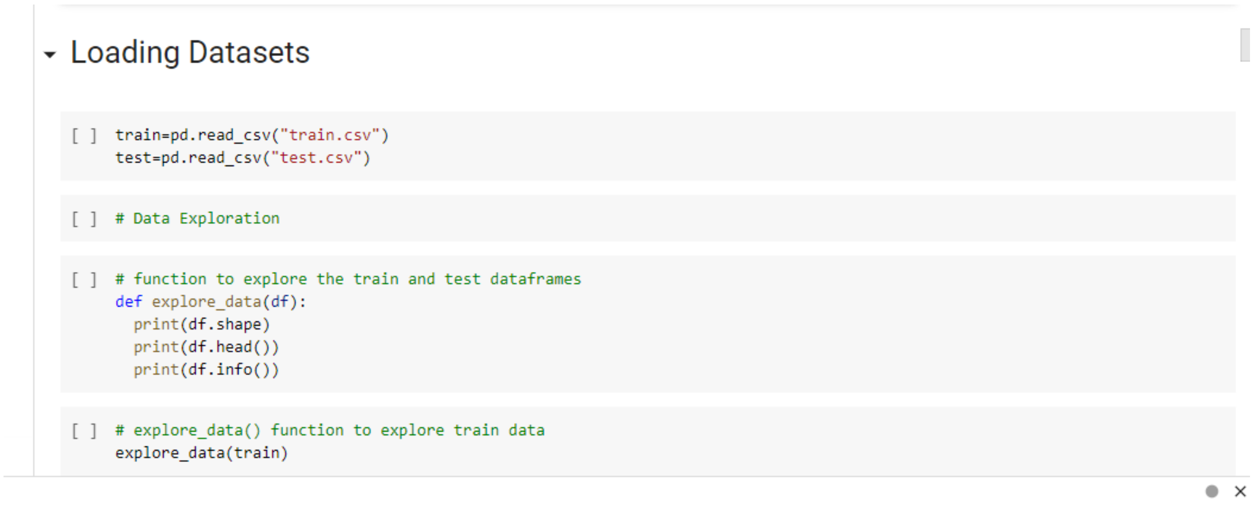
Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

6.1. Imports libraries and Read Data



```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.graph_objects as go
import plotly.express as px
import missingno as msno
import re
import string
import nltk
from nltk.corpus import stopwords
from sklearn import model_selection
from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer
from sklearn.model_selection import train_test_split,GridSearchCV
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import ENGLISH_STOP_WORDS,TfidfTransformer
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.multioutput import MultiOutputClassifier
from sklearn.model_selection import RepeatedStratifiedKFold
from nltk.tokenize import word_tokenize,RegexpTokenizer
```

Loading Data and Applying exploring function to explore the data of train.csv & test.csv: -



Loading Datasets

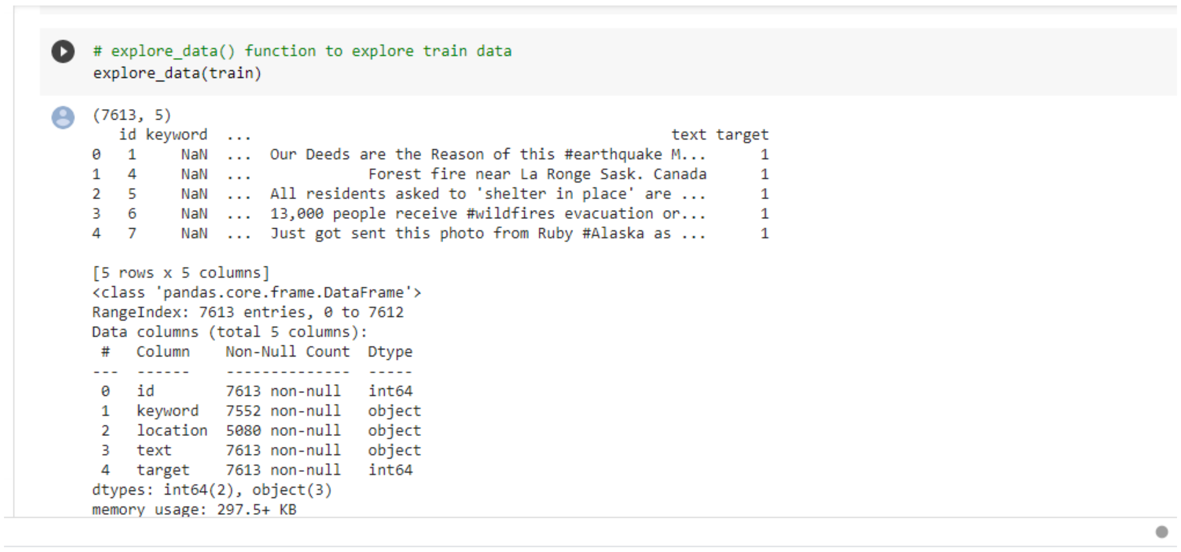
```
[ ] train=pd.read_csv("train.csv")
    test=pd.read_csv("test.csv")

[ ] # Data Exploration

[ ] # function to explore the train and test dataframes
    def explore_data(df):
      print(df.shape)
      print(df.head())
      print(df.info())

[ ] # explore_data() function to explore train data
    explore_data(train)
```

6.2. Glimpse of Data



```
# explore_data() function to explore train data
explore_data(train)
```

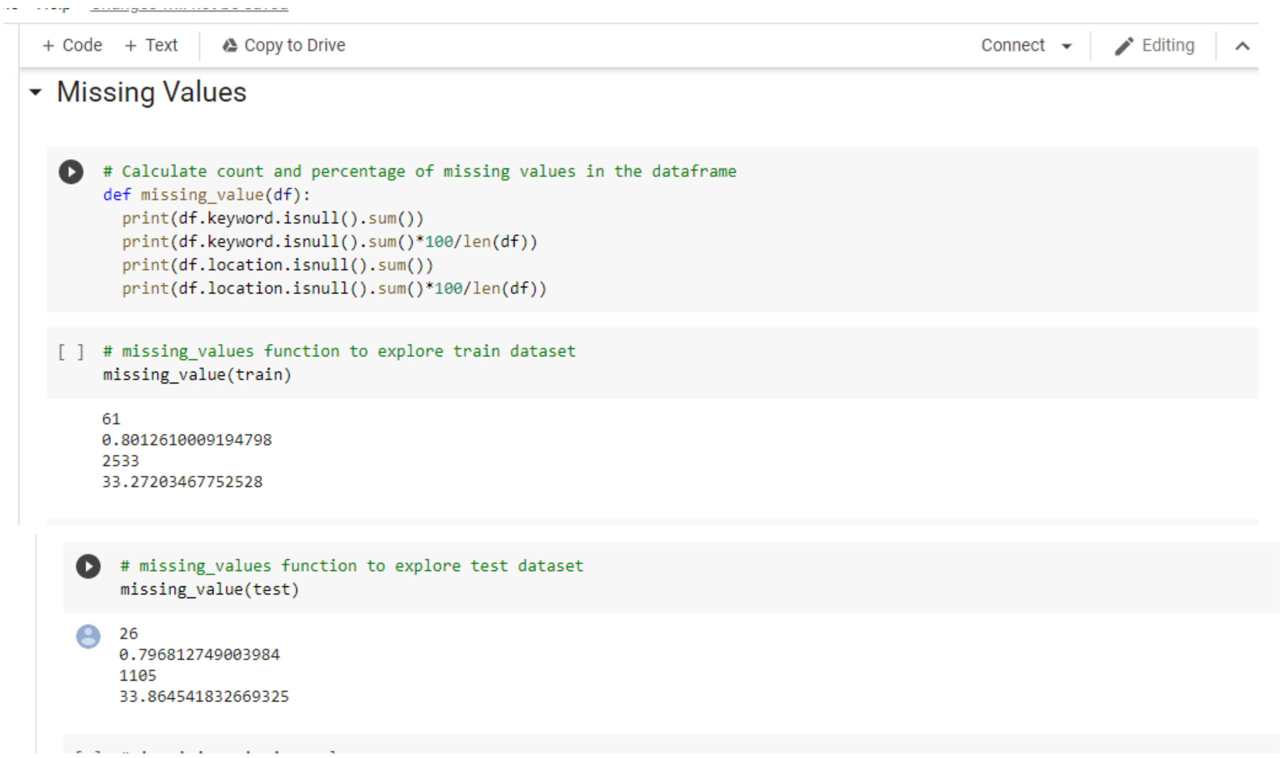
```
(7613, 5)
   id keyword  ...                                               text target
0   1     NaN  ...  Our Deeds are the Reason of this #earthquake M...      1
1   4     NaN  ...             Forest fire near La Ronge Sask. Canada      1
2   5     NaN  ...  All residents asked to 'shelter in place' are ...      1
3   6     NaN  ...  13,000 people receive #wildfires evacuation or...      1
4   7     NaN  ...  Just got sent this photo from Ruby #Alaska as ...      1

[5 rows x 5 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7613 entries, 0 to 7612
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   id        7613 non-null   int64 
 1   keyword   7552 non-null   object
 2   location  5080 non-null   object
 3   text      7613 non-null   object
 4   target    7613 non-null   int64 
dtypes: int64(2), object(3)
memory usage: 297.5+ KB
```

6.3. Statistical Analysis/distribution of target variable

- The statistical analysis section provides crucial information on how the collected data and samples will be analysed to achieve the primary and secondary study aims. The statistical analysis section should have sufficient information for reviewing committees to be able to determine that the methodology is sound and valid for the planned analysis.
- For analysing the data statistically mean median and mode of different variables were calculated based on which average values and range of the variables are determined.

6.4. Examining Missing Values



Missing Values

```
# Calculate count and percentage of missing values in the dataframe
def missing_value(df):
  print(df.keyword.isnull().sum())
  print(df.keyword.isnull().sum()*100/len(df))
  print(df.location.isnull().sum())
  print(df.location.isnull().sum()*100/len(df))
```

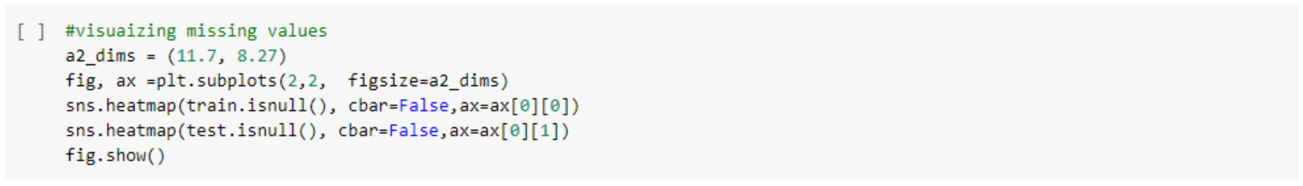
```
# missing_values function to explore train dataset
missing_value(train)
```

```
61
0.8012610009194798
2533
33.27203467752528
```

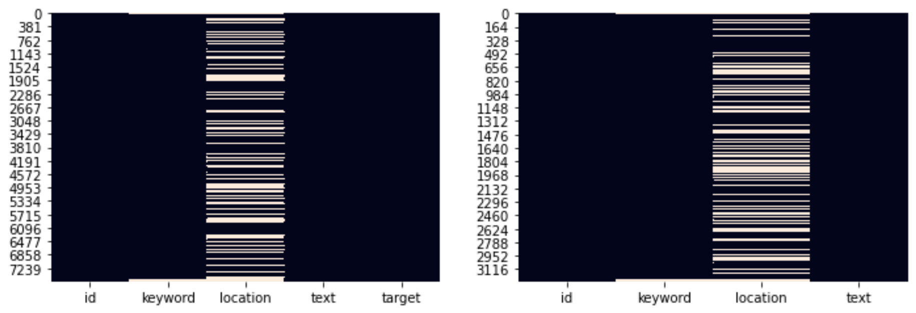
```
# missing_values function to explore test dataset
missing_value(test)
```

```
26
0.796812749003984
1105
33.864541832669325
```

- Visualising the missing values in test.csv and train.csv



```
#visuaizing missing values
a2_dims = (11.7, 8.27)
fig, ax =plt.subplots(2,2,  figsize=a2_dims)
sns.heatmap(train.isnull(), cbar=False,ax=ax[0][0])
sns.heatmap(test.isnull(), cbar=False,ax=ax[0][1])
fig.show()
```

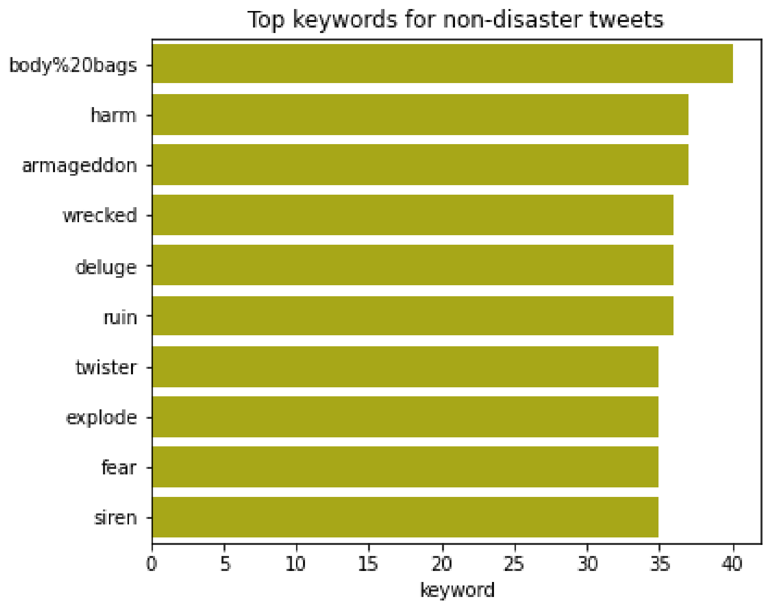


6.5. Column Types

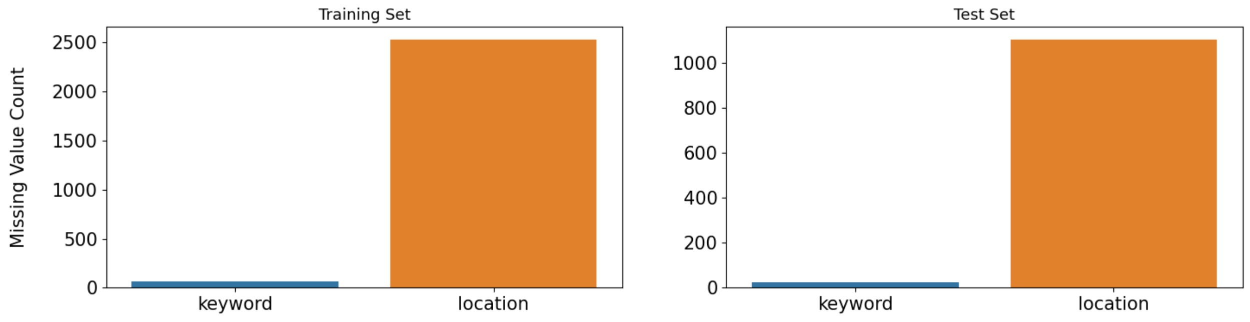
```
 0   id        3263 non-null   int64
 1   keyword   3237 non-null   object
 2   location 2158 non-null   object
 3   text      3263 non-null   object
dtypes: int64(1), object (3)
```

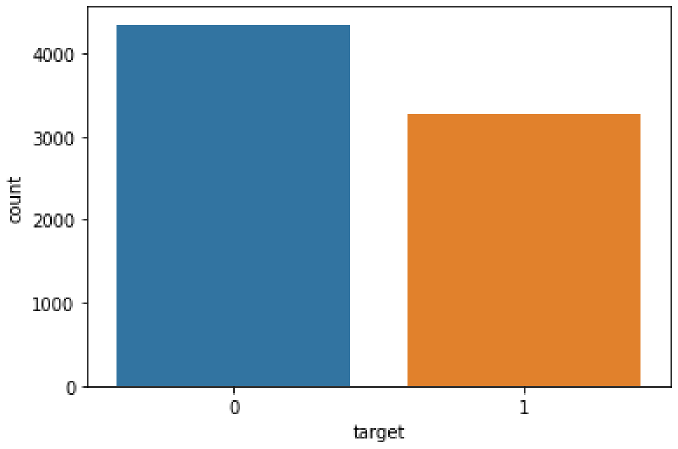
Visualisations: -

Plot showing mostly used keywords of non-disaster tweets: -



Plot showing missing values in location column: -





7. Model Building: -

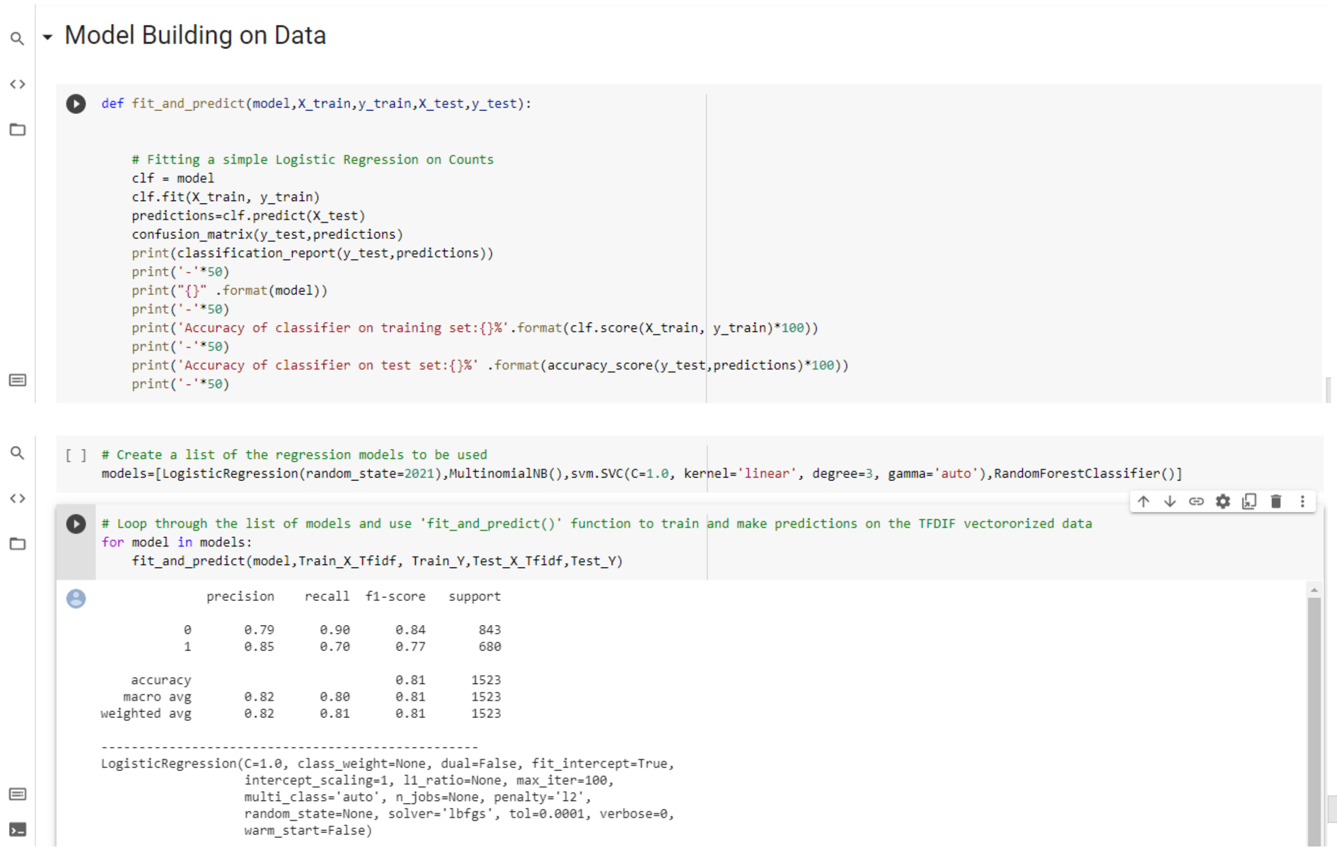
We create a simple classification model using commonly used classification algorithms and other models like random forest, logistic regression, Multinomial NB, XGboost, etc. Then we check the model performances.

We Split the ‘train’ data set into train and ‘test’ data for fitting the model using TF-IDF vectorization.

Results

Based on the training and predictions made by various classifier models, and print the accuracy metrics of the models to summarize the results.

[also telling about the best algorithm with more accuracy]



Model Building on Data

```
def fit_and_predict(model,X_train,y_train,X_test,y_test):


    # Fitting a simple Logistic Regression on Counts
    clf = model
    clf.fit(X_train, y_train)
    predictions=clf.predict(X_test)
    confusion_matrix(y_test,predictions)
    print(classification_report(y_test,predictions))
    print('-'*50)
    print("{}" .format(model))
    print('-'*50)
    print('Accuracy of classifier on training set:{}%'.format(clf.score(X_train, y_train)*100))
    print('-'*50)
    print('Accuracy of classifier on test set:{}%' .format(accuracy_score(y_test,predictions)*100))
    print('-'*50)
```

```
[ ] # Create a list of the regression models to be used
    models=[LogisticRegression(random_state=2021),MultinomialNB(),svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto'),RandomForestClassifier()]
```

```
# Loop through the list of models and use 'fit_and_predict()' function to train and make predictions on the TFDIF vectororized data
for model in models:
    fit_and_predict(model,Train_X_Tfidf, Train_Y,Test_X_Tfidf,Test_Y)
```

```
              precision    recall  f1-score   support

           0       0.79      0.90      0.84       843
           1       0.85      0.70      0.77       680

    accuracy                           0.81      1523
   macro avg       0.82      0.80      0.81      1523
weighted avg       0.82      0.81      0.81      1523

--------------------------------------------------
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

8. Performance Metrics

- After implementing a model and getting some output in forms of a probability or a class, the next step is to find out how effective is the model based on some metric using test data sets. Different performance metrics are used to evaluate different Machine Learning Algorithms.

This problem is a classification problem hence I have calculated confusion matrix, F1-score and accuracy as a performance metric to evaluate model performance.

1.Confusion Matrix:

- The confusion matrix, is a table with two dimensions (“Actual” and “Predicted”), and sets of “classes” in both dimensions. Our Actual classifications are columns and Predicted ones are Rows.
-

n = total predictions	Actual: No	Actual: Yes
Predicted: No	True Negative	False Positive
Predicted: Yes	False Negative	True Positive

Classification Accuracy: It is one of the important parameters to determine the accuracy of the classification problems. It defines how often the model predicts the correct output. It can be calculated as the ratio of the number of correct predictions made by the classifier to all number of predictions made by the classifiers. The formula is given below:

$$\text{Accuracy}=\frac{TP+TN}{TP+FP+FN+TN}$$

Precision: It can be defined as the number of correct outputs provided by the model or out of all positive classes that have predicted correctly by the model, how many of them were actually true. It can be calculated using the below formula:

$$\text{Precision}= \frac{TP}{TP+FP}$$

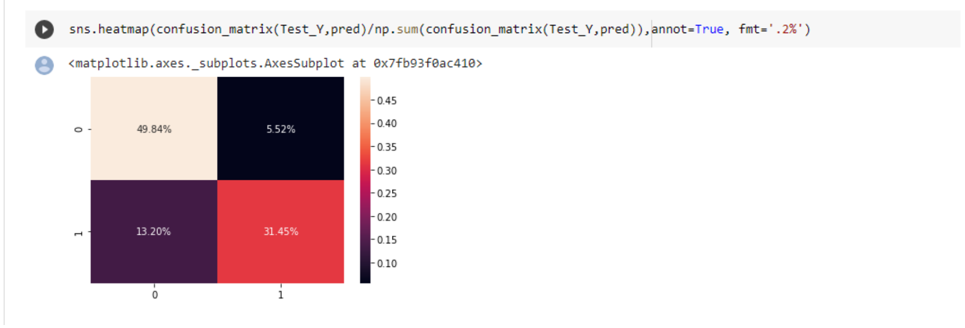
F-measure: If two models have low precision and high recall or vice versa, it is difficult to compare these models. So, for this purpose, we can use F-score. This score helps us to evaluate the recall and precision at the same time. The F-score is maximum if the recall is equal to the precision. It can be calculated using the below formula:

$$\text{F-measure} = \frac{2*Recall*Precision}{Recall+Precision}$$

Need for Confusion Matrix in Machine learning

- It evaluates the performance of the classification models, when they make predictions on test data, and tells how good our classification model is.
- It not only tells the error made by the classifiers but also the type of errors such as it is either type-I or type-II error.
- With the help of the confusion matrix, we can calculate the different parameters for the model, such as accuracy, precision, etc.

- The Confusion matrix in itself is not a performance measure as such, but almost all of the performance metrics are based on Confusion Matrix and the numbers inside it.
  Here, we find the confusion matrix for y_pred, test: -



```
sns.heatmap(confusion_matrix(Test_Y,pred)/np.sum(confusion_matrix(Test_Y,pred)),annot=True, fmt='.2%')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb93f0ac410>
```

2. F1-Score:

- F1-score combines precision and recall relative to a specific positive class. The F1-score can be interpreted as a weighted average of the precision and recall, where an F1-score reaches its best value at 1 and worst at 0.
- Submissions are evaluated using F1 between the predicted and expected answers.
- F1 is calculated as follows:

F_1 = 2 * /frac {precision * recall} {precision + recall}

Where

precision = /frac {TP} {TP + FP}

recall = /frac {TP} {TP + FN}

True Positive = your prediction is 1, and the ground truth is also 1 - you predicted a positive and that's true

False Positive = your prediction is 1, and the ground truth is 0 - you predicted a positive, and that's false

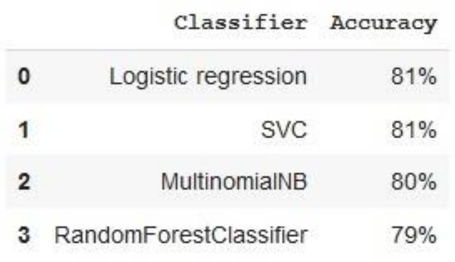
False Negative = your prediction is 0, and the ground truth is 1 - you predicted a negative, and that's false.

3. Accuracy:

- Accuracy in classification problems is the number of correct predictions made by the model over all predictions made.
- Accuracy is good measure when target class variable in the training data set is nearly balanced.

9.Results:

From the above applied machine learning model, logistic regression has the best accuracy 81%.



	Classifier	Accuracy
0	Logistic regression	81%
1	SVC	81%
2	MultinomialNB	80%
3	RandomForestClassifier	79%

10. Challenges and Complications

Working on this data science capstone project went pretty smoothly without many issues. Even though steps in developing this project was smooth but following complications were faced:

- We took the NLP challenge for our project and this area is new and challenging for us since we had to start from scratch.

- New topic and less time also restricted to test few more models from deep neural network area.

11. Future Scope: -

1. Trying neural network and deep neural network models such as RNN (Recurrent Neural Network) might result in higher accuracy and better performance

2. Analysis of people’s sentiment during a disaster by applying sentiment-analysis on the tweet content.

3. Involve the use of other popular social networking sites such as Facebook and Google+ in order to increase our database.

12. Conclusion: -

Machine learning (ML) methods has recently contributed very well in the advancement of the prediction models used for energy consumption. Such models highly improve the accuracy, robustness, and precision and the generalization ability of the conventional time series forecasting tools.

First, we have analysed and explored all the provided tweets data to visualize the statistical and other properties of the presented data. Next, we have performed some exploratory analysis on the data to check type of the data, whether there are unwanted features or if features have missing data. Based on the analysis, we decided to drop 'location' column because it has the most missing data and really have no effect on classification of tweets. The 'text' columns are all text data along with alpha numeric, special characters and embedded URLs. The 'text' column data needs to be cleaned and pre-processed and vectorized before it uses with a machine learning algorithm for the classification of the tweets.

After pre-processing the train and test data, the data was vectorized using and TF-IDF vectorization algorithm and then it was splitted into train and test data, and then various classifiers were fit on the data and predictions were made. Out of all classifiers tested, Logistic Regression () performed the best with the accuracy of 81%.

Finally, Logistic Regression () classifier with default parameters was selected as final model for this NLP classifications challenge and predictions were made on the tweets by live streaming.

13. References

Dataset: -

https://towardsdatascience.com/

https://www.javatpoint.com/

https://plotly.com/dash/

https://www.analyticsvidhya.com/